

1 Ilk Uchrashuv

1.1 Programma nima?

Programma bu aniq bir muammoni yechish uchun ko'rsatmalar ketma ketligidir. Programmist uchun eng muhim kerak bo'ladigan mahorat bu berilgan muammoni hal qilish ketma-ketligini o'ylab topish. Yechimni aniq va to'g'ri bo'lishligi lozim. Qizig'i shundaki, programma tuzishni o'rganish davomida muammoni hal qilish mahoratini sizda shakllanadi. Qancha ko'p programma tuzsangiz shuncha yaxshi, ko'p o'qish yoki kino ko'rish bilan emas :).

Programma oddiydan juda murakkab bo'lishi mumkin: masalan sonlarni qo'shish yoki xujjatdan kerakli so'zlarni qidirish yoki rasm va videolarga ishlov berish va hokazo. Programma quyidagi oddiy ko'rsatmalardan tashkil topishi mumkin:

- Kiritish: klaviaturadan, fayldan, vahokazo.
- Chiqarish: Malumotlarni ekranga chiqarish, faylga saqlash, vahokazo.
- Hisoblash: Matematik amallarni bajarish. Qo'shish va ayirishga vahokazo.
- Tekshirish: Holatlarni tekshirish va to'g'ri kodni ishlatish.
- Takrorlash: Biron-bir harakatni qayta-qayta qilish. Odatda, har safar qisman o'zgarish bilan.

Quvonarlisi shundaki, deyarli shu ko'rsatmalar siz ishlatgan yoki siz qilmoqchi bo'lgan programmalarda bo'ladi. O'shaning uchun, programmalashni biz katta topshiriqni kichikdan kichik topshiriqlarga shunday bo'lamizki – kichik topshiriqlar yuqorida ko'rsatilgan oddiy ko'rsatmalar bilan hal qilinadigan bo'lsin.

1.2 Pythonni ishlatish

Hozirda har bir operatsion sistemada python **interpretor** mavjud. Kompyuteringizdagi terminalni ishga tushiring va python deb yozingda Enter ni bosing. Va quyidagini ko'rishingiz mumkin.

```
1 Python 3.8.5 (default, Jan 27 2021, 15:41:15)
2 [GCC 9.3.0] on linux
3 Type "help", "copyright", "credits" or "license" for more information.
4 >>>
```

>>> belgisi 'prompt' deb ataladi va u python interpretatori tayyorligini bildiradi. Endi programma tuzishga marhamat!

1.3 Birinchi programma

Qaysi programmalash tilini olmag, yoki kitobni o'qimang deyarli hammasida eng birinchi programma "Hello, World" bo'ladi. Bu pythonda quyidagicha bo'ladi:

```
1 >>> print("Hello, World!")
```

Bu **print statementiga** misol. Bunda programma natijani ekranga chiqarayapti, yani:

```
1 Hello, World!
```

Print statementisiz ham qilishimiz mumkin, yani

```
1 >>> "Hello, World!"  
2 'Hello, World!'
```

1.4 Oddiy arifmetika

Pythonda ko'p operatorlar bor. Operatorlar bu mahsus belgilar bo'lib ular turli xil vazifalarni qilishda ishlatiladi. Arifmetik amallar uchun quyidagi operatorlar mavjud: - qo'shish +, - ayirish -, - ko'paytirish *, - bo'lish / - darajaga oshirish **

Pythonda misollar

```
1 >>> 2+2  
2 4  
3 >>> 6-2  
4 4  
5 >>> 2*2  
6 4  
7 >>> 4/5  
8 0.8  
9 >>> 3**2 + 6  
10 15
```

1.5 Qiymat va turlar

Qiymat bu programma bilan ishlaydigan muhim bo'gin. Hozirgacha biz bazi qiymatlarni ko'rib o'tdik: "Hello, World", 4, va 0.8. Bu qiymatlar bir necha turga (class ingliz tilida) tegishli: "Hello, World!" bu Stringga mansub. String bu ketma-ket kelgan harf yoki belgilardan tashkil topgan to'plamdir. 4 bu butun

songa tegishli (integer). 0.8 bu haqiqiy son (floating point number) ga tegishli. Agar siz biron qiymatning qaysi turga tegishligini bilmoqchi bo'lsangiz quyidagicha qilasiz:

```
1 >>> type(4)
2 <class 'int'>
3 >>> type(0.8)
4 <class 'float'>
5 >>> type("Hello, World")
6 <class 'str'>
```

Natijalarga qarasak, 4 ga int ko'ratilayapti bu butun son demakdir (integer ingliz tilida). Bu yerda **class** tur degani. 0.8 floatni ko'rsatayapti yani float point number. 'Hello, World!' str turiga (class) tegishli. str string ning qisqartirilgani.

Etibor beringki biz qiymatlarni quyidagicha yozishimiz ham mumkin '4' yoki '0.8'. Bular songa o'xshaydi lekin ular qo'shtirnoq ichiga yozilgan o'shaning uchun ular string turiga mansub.

```
1 >>> type('4')
2 <class 'str'>
3 >>> type('0.8')
4 <class 'str'>
```

Qolaversa quyidagicha ifodani yozishimiz noto'g'ri

```
1 >>> "Hello, World!" + 1
```

Biz amallarni bajarayotganimizda, hamishi ularning turlari bir-biriga munosib bo'lishligi darkor. Boshqa misol:

```
1 >>> "Hello, " + "World!"
```

1.6 Ona va rasmiy til

Ona tilimiz o'zbek hisoblanadi, yoki inglizlarning ona tilisi ingliz tilidir. Bu tillar insonlar tomonidan yasalmagan, balki vaqt o'tishi bilan shakllanib kelgan.

Rasmiy tillar esa insonlar tomonidan maxsus maqsadlar uchun o'ylab topilgan. Masalan, yo'l harakati uchun til mavjud, yani yo'lda ketayotganda biz har xil belgilarni ko'ramiz. Ularni o'zbekistonning qaysi joyiga bormang bir xil. O'zbek, ingliz yoki tojikga bo'lsin hamma haydovchi uchun tushunarli. Bu yo'l tili insonlar tomonidan malum meyorlar asosida ishlab chiqilgan. Va yo'ldagi holatlarni haydovchiga bildirib ('aytib') turadi.

Xuddi shunga o'xshash python ham rasmiy til bo'lib u insonlar tomonidan ishlab chiqilgan – kompyuterdagi amallarni ifodalash uchun xizmat qiladi

Rasmiy tillar qattiq qoidalarga ega. Masalan tezlikni *10* dan oshirmang degan joyga *100* deb yozib qo'ysak, nima bo'lishligini bilish qiyin emas! – avariya yoki mudhish hodisa ro'y berishi mumkin. Ona tilida gapirilgan qat'iy qoidalar o'ynamaydi va xato qilinsa tezda to'g'irlab ketaverishimiz mumkin. Yoki kerakli ma'lumotni qaytadan takrorlash qo'shimcha qo'shish mumkin. Python tilida har bir qism aniq bo'lishi kerak, kerakli so'zlarni to'g'ri yozish nuqta va vergullarni to'g'ri yozish talab qilinadi aks holda programma ishlamaydi.

1.7 Debugging

Programmistlar xato qilishadi. Xatolarni topib va ularni yechishni debugging (dibagging deb o'qiladi) deb ataymiz.

Masalan quyidagi programma ishlatsangiz:

```
1 >>> print2)
```

Ishlamaydi, va quyidagi xato chiqariladi

```
1 File "<stdin>", line 1
2   print2)
3       ^
4 SyntaxError: unmatched ') '
5 >>>
```

SyntaxError deb ko'rsatilayapti. Yani, biz etiborga bersak, bitta qavsni tashlab ketganligimizni bilishimiz mumkin. Va, aslida programma quyidagiga bo'lishligi kerak edi.

```
1 >>> print(2)
```

Debugging qilish uchun ham malaka kerak, vaqt o'tishi bilan programma qilish davomida o'rganib ketaveramiz. Qolaversa, debugging programmalashdan tashqarida ham foydali bo'lsa ajabmas.

1.8 Foydali terminlar

- Muammo yechish: Bu muammoni ifodalab unga aniq yechim topishdir. Bu ingliz tilida problem solving deb ataladi.

- Programma: Berilgan aniq hisobni amalga oshirish uchun kerak buladigan amallar ketma-ketligi.
- integer: butun sonlarni bildiradigan tur: 12, 0, -50.
- floating-point: ratsional/irratsinal sonlarni bidiruvchi tur: 0.5, 3.143454.
- string: belgilar ketma ketligini ifodalash uchun tur: "Hello, World".
- Syntax: programmada ifodalar yozilishining umumiy qoidalari.
- bug: programmadagi hato bug (inglizchada) deb ataladi
- debugging: xatolarni topish va ularni to'g'irlash.
- print statement: python interpretatoriga ko'rsatma, yani qiymatni ekranga chiqarish uchun
- Interpretator: bu programma boshqa programmani olib uni ishlatadi.
- prompt: interpretator >>> belgisini chiqaradi va uning tayyorligini bildiradi.

1.9 Problem solving

Yuqoridagi programmalarini o'zingiz yana qilib ko'ring, bu safar o'zgarishlar bilan. 1. print statementga oid. Qavslardan birini oling va programma ishlatiga (Enterni bosaiz). Hamma qavsni olib tashlasakchi?

2. print statementga oid. Qo'shtirnoqlardan birini oling va programmani ishlatiga.
3. Mana shunday qilib yozsak programmaga nima bo'ladi 4+-4? 4++4 chi?
4. python interpretatoriga 1 deb yozing va Enterni bosing. Keyin, 01? 0099? Nima bo'ldi?
5. 2+2 ni ishlatiga. 2 2 ni ishlatiga. Nima bo'ldi?
6. "Hello, World!" + 1 ni ishlatiga? ishlamasa nimaga ishlamaganligini ayting

Python interpretatorini ishlatigda quyidagilarni bajaring: 1. 6 soat 30 sekundni sekundga aylantiring. 2. 2 ni 10 darajasini 555 ga bo'ling. 3. Ekranga "Salom, Dunyo" deb chiqaring. 4. Ekranga "Salom, Dunyo ingliz tilida 'Hello World' degani" deb chiqaring